

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

NOTIFIKACE POMOCÍ USB HID ZAŘÍZENÍ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

DANIEL JANÁČEK

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

NOTIFIKACE POMOCÍ USB HID ZAŘÍZENÍ

USB HID DEVICE NOTIFICATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

DANIEL JANÁČEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. LADISLAV ŠŤASTNÝ

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Daniel Janáček

ID: 154742

Ročník: 3

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Notifikace pomocí USB HID zařízení

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je realizace rozšíření využívajícího notifikátor v HID standardu. Jeho úlohou bude pomocí zařízení připojeného k PC upozornit uživatele na různé události, které mohou nastat během používání komunikátoru. Zadání lze shrnout do následujících bodů:

1. Zpracujte rešerši možných řešení notifikátorů a seznamte se se standardem HID.
2. Definujte požadavky na notifikátor, včetně požadavků na SW.
3. Vyberte HW řešení ve standardu HID.
4. Navrhněte a realizujte rozšíření pro Mirandu NG využívající vybraný HW.
5. Testujte a vyhodnoťte řešení.

DOPORUČENÁ LITERATURA:

KENT, Jeffrey A. C++ bez předchozích znalostí: [přůvodce pro samouky]. Vyd. 1. Brno: Computer Press, 2009, 254 s. ISBN 978-80-251-2411-6.

Termín zadání: 9.2.2015

Termín odevzdání: 25.5.2015

Vedoucí práce: Ing. Ladislav Šťastný

Konzultanti bakalářské práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Cieľom tejto práce je zoznámenie sa so štandardom HID, ktorý využívajú externé USB zariadenia, ako napríklad notifikátory. Ďalšou úlohou je vybrať takéto hardvérové zariadenie a následne ku nemu naprogramovať plugin, spolupracujúci s klientom Miranda NG. Toto zariadenie bude slúžiť hlavne na informovanie používateľa o prichádzajúcich správach, emailoch a iných udalostiach. Následne sa popisuje vytvorenie softvérového doplnku k tomuto zariadeniu, jeho dizajn, popis jednotlivých funkcií a nakoniec test funkčnosti pluginu.

KĽÚČOVÉ SLOVÁ

HID, USB notifikátor, Miranda NG

ABSTRACT

The objective of the thesis is familiarization with standard HID that is utilized by external USB devices, such as notifiers. Moreover, another objective consists in choosing such a hardware device and in programming plugin collaborating with Miranda NG client. This device will mainly be instrumental in informing the user about incoming messages, emails and other events. Subsequently, it is defined creation of software additional item for the device, its design, description of particular functions and test concerning functionality of plugin.

KEYWORDS

HID, USB notifier, Miranda NG

JANÁČEK, Daniel *Notifikace pomocí USB HID zařízení*: bakalárska práca. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2015. 41 s. Vedúci práce bol Ing. Ladislav Šťastný

PREHLÁSENIE

Prehlasujem, že som svoju bakalársku prácu na tému „Notifikace pomocí USB HID zařízení“ vypracoval samostatne pod vedením vedúceho bakalárskej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/nebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., o právu autorskom, o právach súvisejúcich s právom autorským a o zmene niektorých zákonov (autorský zákon), vo znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka č. 40/2009 Sb.

Brno

.....

(podpis autora)

POĎAKOVANIE

Rád by som poďakoval vedúcemu bakalárskej práce pánovi Ing. Ladislavovi Šťastnému za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

Brno

.....

(podpis autora)

OBSAH

Úvod	10
1 Rozbor zadania	11
1.1 Cieľ práce	11
1.2 USB notifikátor	11
1.3 Dostupné zariadenia na trhu	11
1.3.1 Razer DeathStalker Ultimate	11
1.3.2 Sony PS DualShock 4	12
1.3.3 Blink	13
1.3.4 USB Real time notification device	14
1.4 Human interface device	15
1.4.1 Descriptor	15
1.4.2 HID Class	17
1.4.3 Boot protocol	18
2 Požiadavky na zariadenie a softvér	19
2.1 Miranda NG	19
2.1.1 Podporované protokoly	20
2.2 TortoiseSVN	21
3 Výber zariadenia	22
3.1 Dream Cheeky 815 Webmail Notifier	22
4 Návrh obslužného pluginu	24
4.1 Uživatelské prostredie	24
4.1.1 Nastavenia	24
4.1.2 Pravidlá	26
5 Vývoj a tvorba pluginu	28
5.1 Popis komunikácie s notifikátorom	28
5.1.1 PID a VID Identifikátory	28
5.2 Zaznamenanie udalostí	29
5.3 Popis zdrojových kódov pluginu	31
5.3.1 Princíp notifikácie	32
5.3.2 Použité zdrojové kódy	32
5.3.3 Metódy starajúce sa o výslednú notifikáciu	33
6 Testovanie pluginu	34

7 Záver	36
Literatúra	37
Zoznam symbolov, veličín a skratiek	39
Zoznam príloh	40
A Priložené CD	41
A.1 Zdrojové kódy pluginu	41

ZOZNAM OBRÁZKOV

1.1	Razer DeathStalker Ultimate [5]	12
1.2	Sony PS DualShock 4 [9]	13
1.3	Blink [1]	13
1.4	USB Real time notification device [11]	14
1.5	Typické HID zariadenia [7]	15
1.6	Stromová štruktúra descriptoru[13]	16
2.1	TortoiseSVN	21
3.1	Dream Cheeky 815 Webmail Notifier [2]	22
3.2	Pohľad do vnútra	23
4.1	Úžívateľské prostredie - Nastavenia	26
4.2	Úžívateľské prostredie - Pravidlá	27
5.1	Identifikátory PID a VID	28
5.2	Vývojový diagram	30
6.1	Test notifikátora	34

ZOZNAM TABULIEK

5.1	Tabuľka príkazov	29
-----	----------------------------	----

ÚVOD

V súčasnosti vo svete plnom chaosu a uponáhľaného spôsobu života, chce byť každý o všetkom čo najrýchlejšie informovaný, aby nepremeškal nejakú dôležitú udalosť, správu alebo dôležitý email. Preto sa vymýšľajú rôzne druhy zariadení, ktoré sa snažia tomuto zabrániť tým, že užívateľa nejakým spôsobom ihneď upozornia.

Začiatok práce obsahuje popis toho čo vlastne USB notifikátor je a aký význam má v modernej dobe počítačov. Ďalej sa venuje rozboru štandardu HID a jeho úlohe v zariadeniach obsahujúcich USB port, popis triedy HID class ako aj Boot protokolu. Taktiež táto práca obsahuje dostupné riešenia na trhu, popisuje softvérové a hardvérové požiadavky, a dôvod prečo bolo vybrané určité jedno zariadenie.

V jadre práce je navrhnutý celkový dizajn, naprogramované vlastnosti pluginu notifikátora, a popis použitých metód pre klienta Miranda NG. Ďalej popisuje komunikáciu medzi pluginom a klientom, a spôsob, akým plugin oboznamuje užívateľa o novej udalosti.

Na záver je v poslednej kapitole, v krátkosti popísané testovanie pluginu, to ako sa testovalo a čo bolo k tomu potrebné. Toto testovanie je nevyhnutné hlavne počas odlaďavania softvérovej časti, hľadania chýb a dosiahnutia výslednej verzie.

1 ROZBOR ZADANIA

1.1 Cieľ práce

Cieľom tejto práce je oboznámiť sa so štandardom HID, ktorý je využívaný v zariadeniach pripojených k počítači cez port USB, následne ku nemu navrhnúť softvér pre notifikačné zariadenie a komunikovať s programom Miranda NG. Softvérová časť bude obsahovať plugin, implementovaný priamo do prostredia programu, ktorý bude schopný komunikovať a odosielať príkazy cez USB port do notifikátoru a následne bude vykonávať naprogramované operácie. Ako vývojové prostredie bude použité Visual Studio 2013. Použitý programovací jazyk bude C++. Hardvérová časť bude pozostávať z RGB LED diódy. Elektronika bude využívať napájanie z USB portu.

1.2 USB notifikátor

USB notifikátor je zariadenie, ktoré sa objavilo hlavne spolu s postupným rozširovaním instant messagingu a sociálnych sietí. Ide vždy o nejaký druh elektroniky, ktorá má za cieľ informovať užívateľa o nejakej novej práve vykonanej zmene, udalosti alebo správe. Môže existovať vo viacerých podobách. Komerčne dostupné riešenia väčšinou zahŕňujú upozorňovanie na nejakú udalosť do svojich už stávajúcich produktov. Postupným rozširovaním, čoraz viac už bežne dostupných, herných konzol, sa snažia výrobcovia nejakým odlišiť od ostatných a preto pre užívateľov vymýšľajú a pridávajú stále ďalšie a ďalšie vychytávky do svojich herných ovládačov.

1.3 Dostupné zariadenia na trhu

V nasledujúcich podkapitolách sú popísané dostupné notifikačné zariadenia na trhu.

1.3.1 Razer DeathStalker Ultimate

Firma Razer, známy výrobca komponentov pre hráčov počítačových hier, vyrába klávesnice s dotykovým LCD displejom. Hlavnou výhodou je, že ak hráč práve hrá na počítači hru a napíše mu nejaký kamarát správu, môže mu jednoducho odpovedať bez toho, aby vypol rozohranú hru. Jednou z nich je Razer DeathStalker Ultimate. Obsahuje štvorpalcový dotykový displej, ktorý dokáže zobrazíť správy z obľúbených internetových služieb ako sú napríklad Facebook, GMail, Twitter alebo taktiež videá z Youtube. Všetky udalosti sa tak dajú počas hry pohodlne vybaviť, bez prerušenia hry. Doinštalovaním ďalších takzvaných widgetov je možné využívať doplnkové

vlastnosti tejto klávesnice aj inými spôsobmi.

Vzhľadom k tomu, že klávesnica má podobné rozmery ako akákoľvek iná klávesnica, museli návrhári vymyslieť kam tento displej umiestniť. Preto sa na tejto klávesnici nenachádza numerický blok, ktorý je práve nahradený dotykovým displejom. Takže numerické čísla v tejto časti sa tu nachádzajú v podobe bez hmatovej odozvy a sú len softvérovo meniteľné. Ďalej sa táto plocha dá použiť ako touchpad s podporou gest. Do vstavanej pamäte si môžete ukladať vami nadefinované makrá pre programovateľné klávesy a tak jednoducho trebárs odoslať vami nadefinovaný súvislý text jedným stlačením tlačidla. O všetky tieto vlastnosti sa stará softvér Razer Synapse, ktorý sa dokáže sám aktualizovať a stiahnuť nové ovládače a firmvér.[5]



Obr. 1.1: Razer DeathStalker Ultimate [5]

1.3.2 Sony PS DualShock 4

Ďalšie zariadenia, ktoré sú zhodou okolností znovu zamerané na hráčov počítačových hier sú Gamepady. Obsahujú taktiež ďalšie možnosti ako oznámiť, že sa udialo niečo nové. Presným príkladom je herný ovládač od firmy SONY pre platformu Playstation. Má v sebe integrovanú vibračnú odozvu a LED ukazovatele, ktoré upozorňujú hráča v prípade nečakanej akcie. Taktiež má malý touchpad, pomocou ktorého sa dá vyriešiť väčšina bežných udalostí. Medzi jeho ďalšie súčasti patrí aj reproduktor, ktorý vás dokáže ešte viac vtiahnuť do deja hrania napríklad aj tým, ak vám v hre priamo počas hrania začne vydávať zvuky policajnej vysielacky alebo mobilného telefónu postavičky z hry. Obsahuje aj mnoho ďalších programovateľných tlačidiel. Taktiež aj tlačidlo pre zdieľanie záberov z hrania, ktoré ich následne odošle priamo

na sociálne siete. Takto sa môže hráč okamžite podeliť so svojimi priateľmi o svoje úspechy.[10]



Obr. 1.2: Sony PS DualShock 4 [9]

1.3.3 Blink

Blink je svetelný indikátor, slúžiaci na notifikovanie rôznych udalostí naprieč rôznymi zariadeniami. Neslúži len na konkrétne zameranie ako sú počítače, ale aj pre servery v rackoch, kde môže notifikovať rôzne chyby, upozornenia alebo oznamovať nové úlohy. Podobne ako aj v nižšie spomenutom notifikátore Dream Cheeky 815 Webmail Notifier 2 aj tento obsahuje RGB LED diódu. Na rozdiel od neho navyše ponúka viacero funkcií. Ponúka teda až 16 miliónov druhov farieb čož je podstatne viac, ďalej dokáže vytvoriť pulzujúce svetlo tým, že sa dá meniť jas jednotlivých farieb LEDky. Výrobca sa zameral hlavne na dizajn, aby tak vzniklo čo najmenšie zariadenie, ktoré moc nevytŕča z portu USB. Podľa stránok výrobcu ho využívajú aj známe spoločnosti ako sú Microsoft, Pixar, Facebook, Adobe a mnoho ďalších.[1]



Obr. 1.3: Blink [1]

1.3.4 USB Real time notification device

Mimo komerčné riešenia od väčších firiem, sa do návrhu vlastného notifikátora pustili aj priaznivci podomácky vyrobených hračiek. Toto zariadenie je podstatne jednoduchšie a obsahuje len jednoduché prvky ako je grafický monochromatický LCD displej, piezo reproduktor a senzor poklepania na zariadenie. Toto zariadenie narozdiel od môjho implementovaného riešenia, používa svoj vlastný softvér, ktorý je naprogramovaný v programovacom jazyku JAVA. Taktiež má svoje vlastné API, čiže je ho možné jednoducho upraviť.

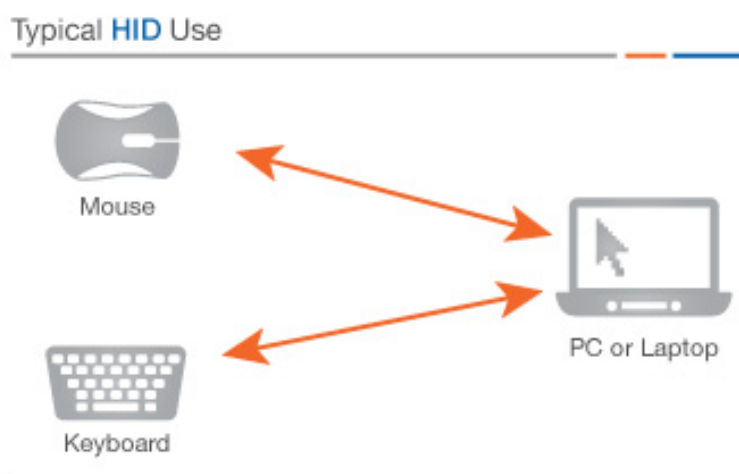
Narozdiel od ostatných masovo rozšírených zariadení, je toto vytlačené na 3D tlačiarňi a to hlavne z dôvodu, nie tak veľkého odbytu. Okrem notifikácií z väčšiny sociálnych sietí ako facebook, twitter a podobne umožňuje zobraziť napríklad aktuálny čas alebo aktuálne počasie, ktoré získava prostredníctvom programu z internetu. Hardvér je postavený na vývojovej doske Arduino Uno. Pozostáva z mikroprocesora Atmega 328 a FTDI čipu, ktorý zabezpečuje komunikáciu medzi počítačom a mikroprocesorom.



Obr. 1.4: USB Real time notification device [11]

1.4 Human interface device

Ako už vyplýva z názvu, tento štandard je určený na interakciu medzi počítačom a človekom. Čiže na začiatku dá človek nejaký podnet, vstup a na konci očakáva nejaký výstup. Jeho hlavnou úlohou je zjednodušiť inštaláciu a následné používanie externých zariadení. S týmto štandardom prišlo až USB, keďže staršie porty ako napríklad PS/2 určené pre myši a klávesnice nepodporovali HID. Dávnejšie boli potrebné špeciálne ovládače zvlášť pre každé zariadenie, dnes vďaka tomuto štandardu dokáže počítač rozpoznať charakter zariadenia či už je to myš alebo klávesnica a ovládače k týmto zariadeniam má ihneď k dispozícii bez akéhokoľvek ďalšieho zásahu používateľa.



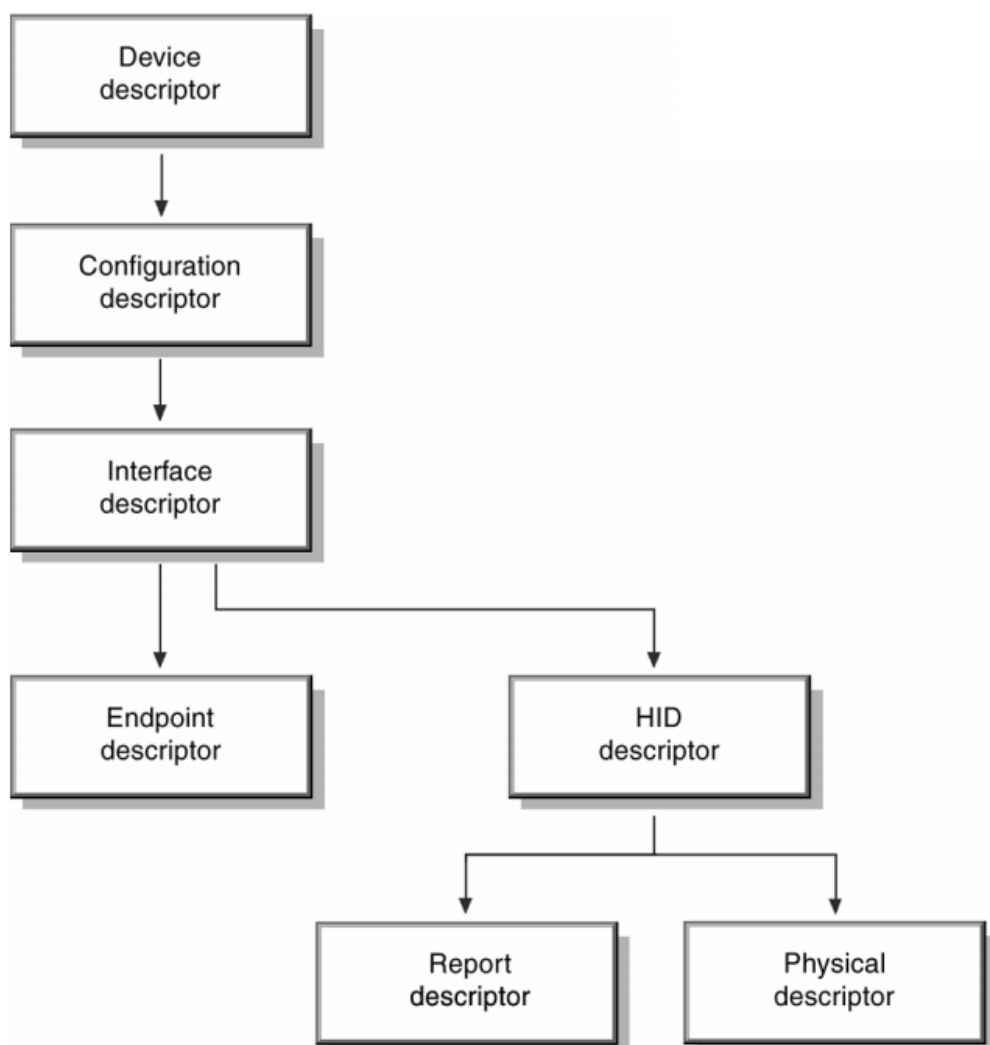
Obr. 1.5: Typické HID zariadenia [7]

Pred tým než HID zariadenie začne komunikovať a prenášať dáta medzi hostiteľom a zariadením, musí byť takzvané správne „enumerované“. Tento proces pozostáva z viacerých volaní takzvaných descriptorov, ktoré sú popísané nižšie. Hostiteľ je vždy nejaké komplexnejšie zariadenie, pretože musí spracovať množstvo dát z descriptoru, napríklad počítač.

1.4.1 Descriptor

Deskriptor si je možné predstaviť ako naprogramované pole bajtov, v ktorom sú uložené informácie o zariadení. Podľa toho, akú informáciu obsahujú sa delia na:

- Device descriptor - obsahuje PID a VID identifikátory, slúžiace na identifikovanie zariadenia
- Configuration descriptor - určuje spôsob napájania, maximálny odber prúdu...
- Interface descriptor - obsahuje informácie o triede, v ktorej bude zariadenie komunikovať - HID, Mass Storage...
- Endpoint descriptor - popisuje smer komunikácie, typ synchronizácie, maximálnu veľkosť paketu...



Obr. 1.6: Stromová štruktúra descriptoru[13]

Po skončení enumerácie je zariadenie pripravené na použitie. Enumerácia sa až do odpojenia zariadenia znovu nevykonáva. Po tomto procese môže začať prenos dát. Všetky dáta musia mať určitú štruktúrovanú formu, ktorá je určená v report descriptor. Physical report obsahuje už len dodatočné dáta, ktoré väčšinou nebý-

vajú definované. Sú to informácie o tom, ako sa zariadenie používa, či je určené pre pravú alebo ľavú ruku, alebo určenie, že dodatočné tlačidlo na boku myši, je pre palec.[14]

Tieto reporty si zariadenie a hostiteľ predávajú pomocou takzvaných „pipes“ alebo rúr dvomi spôsobmi:

- control pipe - zmeny stavu na základe príkazov, zariadenie a hostiteľ si predávajú reporty pomocou žiadostí alebo takzvaných „requests“
 - Get_Report - získanie reportu
 - Set_Report - zapísanie reportu
- interrupt pipe - zmena stavu vykonaná prerušením, získanie dát, ktoré neboli vyžiadané, používa sa hlavne pri klávesniciach, kedy každé stlačenie klávesy vyvolá prerušenie a tak vykonanie nadefinovanej udalosti

Toto všetko zastrešuje trieda HID Class.

1.4.2 HID Class

HID Class alebo taktiež „HID trieda“ je určená na sprístupnenie funkcií pripojeného hardvéru k počítaču, pre ostatné programy. Túto triedu obsahuje každé moderné zariadenie. Používa sa hlavne vtedy, keď zariadenie obsahuje viacero rôznych rozhraní. Napríklad USB telefón pozostáva z tlačidiel, ktoré využívajú HID class a slúchadla s mikrofónom, ktoré využívajú USB audio device class. Funkcie pre túto triedu sa v operačnom systéme Windows nachádzajú v systémovej knižnici pod názvom hid.dll.[12]

1.4.3 Boot protocol

Tak ako to občas býva, môžu mať rôzne operačné systémy, rôzne druhy problémov s pripojenými zariadeniami a hlavne ovládačmi s tým spojenými. Pretože nie každý hosťiteľ si dokáže poradiť s každým zariadením, HID definuje takzvaný „boot protocol“, ktorý špecifikuje len určité podporované vlastnosti zariadenia. S týmto protokolom pracujú dva druhy zariadení:

- Myši
- Klávesnice

Vďaka tomuto protokolu dokážu tieto zariadenia pracovať aj vtedy, ak nemajú načítané svoje ovládače. Typickým príkladom je situácia, kedy je potrebné využívať klávesnicu pre pohyb v nastaveniach BIOSu, ale operačný systém s ovládačmi nie je nabehnutý. Preto súčasťou tohto protokolu pre tieto dve zariadenia je nadefinovaná tabuľka „HID Usage Tables“, ktorá obsahuje funkcie, ktoré dokáže zariadenie vykonať aj bez ovládačov. Pre klávesnicu sú to všetky základné klávesy, okrem tých, ktoré pridávajú výrobcovia do svojich multimediálnych klávesníc. Čo sa týka myši, tá dokáže bez ovládačov zaznamenať jej pohyb a tri tlačidlá, ktoré obsahuje.[14]

Po zhrnutí toho, čo trh ponúka, inšpirácií možností, ktoré dostupné zariadenia dokážu a zoznámením sa s protokolom HID, je potrebné sa posunúť k výberu zariadenia, ktoré bude vhodné pre túto prácu.

2 POŽIADAVKY NA ZARIADENIE A SOFT-VÉR

Zariadenie, ktoré by malo slúžiť ako notifikátor, by malo spĺňať nasledujúce požiadavky:

- Najdôležitejšou vlastnosťou zariadenia musí byť možnosť implementácie do užívateľského prostredia programu Miranda NG.
- V súčasnosti majú všetky počítače USB port, cez ktorý komunikujú s ostatnými externými zariadeniami. Práve preto sa pri výbere zohľadňovalo toto rozhranie. Medzi ďalšie výhody patrí aj napájanie priamo z tohoto portu, čiže nie je potrebné pripájať žiaden doplnkový napájací zdroj.
- Vzhľadom k tomu, že pri práci na počítači nechce byť väčšina ľudí vyrušovaná nejakými akustickými zvukmi, ale zároveň chce byť ihneď informovaná, hľadalo sa zariadenie, ktoré využíva na signalizáciu viacfarebnú RGB LED diódu. Výhodou takéhoto riešenia je aj fakt, že oproti nejakému druhu zvuku, LEDka žiadno nevyrušuje a môže svietiť alebo blikať aj dlhšiu dobu.
- Medzi nezanedbateľné stránky zariadenia patrí aj jeho celkový vzhľad. Keďže hlavným a jediným druhom signalizácie je LED dióda, mal by byť notifikátor vyrobený z nejakého druhu priesvitného materiálu, aby mohlo svetlo prechádzať skrz tento materiál.
- Dostatočná dĺžka USB káblu je potrebná hlavne ak chceme notifikátor pripojiť ku stolnému počítaču a nie notebooku.

2.1 Miranda NG

Je multiprotokolový klient pre operačný systém Windows, ktorý je nástupcom jeho predchodcu Mirandy IM. Je vydávaný pod licenciou GPL General Public License čo znamená, že dovoľuje hocikomu nazrieť do zdrojových kódov programu a študovať ako program funguje. Slúži ako komunikačný nástroj, ktorý zlučuje viaceré služby internetu, ako sú emaily, sociálne siete, jabbre a iné do jedného kompaktného programu.

Vďaka svojej otvorenosti umožňuje komukoľvek stiahnuť si zdrojové kódy z internetu a ľubovoľne si tento program upravovať. Taktiež umožňuje pridávanie rôznych pluginov, či už stiahnutých z internetu, alebo sa dá vytvoriť vlastný podľa seba. Napríklad za zmienku stojí plugin GMailNotifier, ktorý dokáže upozorniť v prípade prichádzajúceho emailu do vašej schránky a takýmto spôsobom je možné získať náhradu za emailového klienta.

Miranda NG umožňuje aj zmenu celkového vzhľadu podľa svojich preferencií. Jednoducho sa tak môže napríklad nastaviť druh a farbu písma alebo pozadia. Nastaviť sa dajú aj klávesové skraty pre zjednodušený prístup k jednotlivým akciám a vyvolať tak jednotlivé ponuky programu alebo zmeniť svoj stav ako napríklad Pripojený/Odpojený a podobne. Všetky nastavenia sú zhromaždené do jednotného dizajnového rozhrania odkiaľ sa sa dajú meniť všetky potrebné parametre. Taktiež je možné medzi pluginmi ľahko vyhľadávať a zobrazovať tak nastavenia od konkrétneho pluginu, ktorý je aktuálne potrebný. Podrobnejšie informácie ako aj možnosť stiahnutia tohto programu je na oficiálnej webovej stránke softvéru. [4]

2.1.1 Podporované protokoly

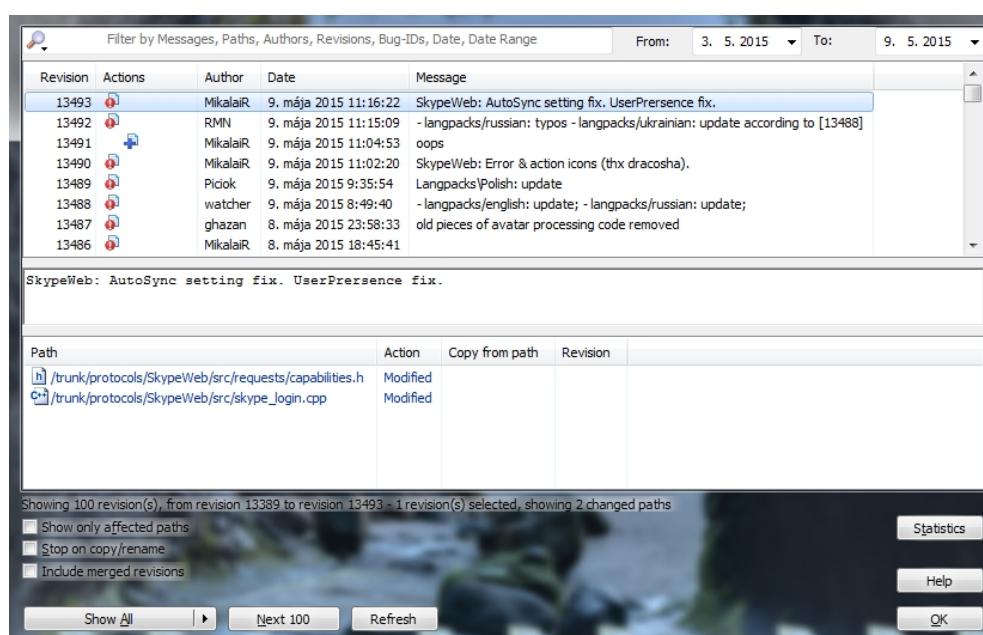
- AIM
- Dummy
- Em-LAN
- Facebook
- Gadu-Gadu
- ICQ
- IRC
- Jabber
- MRA
- MSN
- Omegle
- Sametime
- Tlen
- Twitter
- VKontakte
- WhatsApp
- Xfire
- Yahoo

2.2 TortoiseSVN

Keďže vytváranie programu Miranda NG a pluginov s ním spätých je založené na tom, že ktokoľvek sa môže zapojiť do vývoju, je potrebné zabezpečiť nejaký spôsob, ako bude možné si zadovážiť zdrojové kódy tohto programu.

Na toto slúži verzovací nástroj, taktiež zvaný „Apache Subversion klient“, ktorý je priamo určený na vývoj softvéru. Hlavnou úlohou týchto klientov je zabezpečiť možnosť stiahnutia ktorejkoľvek verzie zdrojových kódov vyvíjaného programu, ktorú vytvoril napríklad kolega, alebo niekto iný z tímu podieľajúceho sa na jeho tvorbe. Táto výhoda je veľmi dobrá hlavne vtedy, keď niekto bez dôkladného skontrolovania svojich zdrojových kódov nahrá novú revíziu programu, ktorá nefunguje správne a ďalší vývojári s ňou nemôžu pracovať. Takýmto spôsobom sa muselo postupovať aj počas programovania tohto pluginu, kedy nastal vlastne problém, kedy grafika celého programu niekoho chybou zmizla.

Na správu zdrojových kódov a spravovanie nových verzií programu bol použitý program TortoiseSVN, ktorý je jeden z najlepších programov dostupných zadarmo, ktoré ponúka jednoduché užívateľské prostredie a množstvo užitočných funkcií. Sú to napríklad: zakázanie nahrávania súborov s príponami určitých typov. Táto funkcia sa hodí hlavne vtedy, keď kompilátor vytvorí plno dočasných súborov, ktoré môžu zaberať gigabajty miesta a ostatným vývojárom sú tieto súbory zbytočné.



Obr. 2.1: TortoiseSVN

3 VÝBER ZARIADENIA

V súčasnosti sa pracuje na čisto softvérovej stránke notifikátoru, aby neskôr mimo rozsah tejto práce mohol vzniknúť aj nový druh hardvéru, ktorý by bol súčasťou embedded zariadení a obsahoval tak aj ďalšie funkcie, ako napríklad priame zobrazovanie správ na displeji a väčšiu interakciu s používateľom. Mohol by tak byť bežnou súčasťou zariadení v domácnosti, áut alebo rôznych druhov elektroniky. Preto sa na odskúšanie pluginu vybralo jedno z viacerých už hotových zariadení na trhu, ktoré majú svoj vlastný obslužný program.

3.1 Dream Cheeky 815 Webmail Notifier

Zariadenie, ktoré spĺňa hore uvedené vlastnosti sa nazýva Dream Cheeky 815 Webmail Notifier. S počítačom komunikuje cez USB port, z ktorého je zároveň aj napájané. Vo vnútri sa nachádza viacfarebná RGB LED dióda, ktorá dokáže meniť 7 druhov farieb. Taktiež podporuje HID štandard, ktorý zjednodušuje inštaláciu zariadenia na absolútnu nulu.

Keďže pri návrhu pluginu sa počíta s jeho voľným šírením, rozhodlo sa ho programovať pre zariadenie, ktoré dokáže byť masovo rozšírené najmä kvôli jeho dostupnej cene. Pretože väčšina ľudí nechce míňať veľa peňazí na doplnky a podobné gadgety ku počítaču, hľadalo sa zariadenie do limitu 10 dolárov. Limit v mene dolár je práve preto, lebo ak chcete nájsť čo najlacnejšie veci, musíte sa začať obzerať po zahraničných internetových obchodoch najmä z Číny. Ceny sa rôzne líšili. Výsledná cena zariadenia bola 6.96 doláru. Poštovné samozrejme zadarmo.



Obr. 3.1: Dream Cheeky 815 Webmail Notifier [2]



Obr. 3.2: Pohľad do vnútra

Po výbere vhodného hardvéru, je následne potrebné si určiť jeho funkcie, a akým spôsobom bude notifikátor spolupracovať s klientom Miranda NG, ktorý je popísaný v nasledujúcej kapitole.

4 NÁVRH OBSLUŽNÉHO PLUGINU

Celý klient ako aj jeho pluginy sú naprogramované v jazyku C++ a ako vychodzí kompilátor sa používa Visual Studio 2013 v jeho aktuálne najnovšej verzii. Práve preto sa pri vytváraní tohto pluginu tiež použilo rovnaké vývojové prostredie. Pre študentov Fakulty elektrotechniky a komunikačných technológií VUT v Brně ho je možné v rámci projektu MSDNAA stiahnuť zdarma zo stránok fakulty.[8] Pri jeho tvorbe sa bral do úvahy aj celkový vzhľad klienta a v tomto duchu sa vytváralo aj prostredie pluginu tak, aby nejako výrazne nevybočovalo z línií programu.

4.1 Uživatelské prostredie

Prostredie je spravené, čo najviac intuitívne, aby nastavenia boli, čo najjednoduchšie pochopiteľné. Pri každom spustení Mirandy sa načítajú údaje z databázy používateľa. Pri ich neskoršej zmene v nastaveniach pluginu, sa po potvrdení uložia, a teda nie je potrebné opakované nastavovanie. Cez menu pluginu v programe je možné jednoducho meniť správanie a vykonávanie jednotlivých akcií, podľa vlastných preferencií. Menu pluginu je rozdelené do dvoch základných častí:

- Nastavenia
- Pravidlá

4.1.1 Nastavenia

Táto záložka obsahuje zaškrŕavacie políčka pre udalosti, na ktoré má notifikátor zareagovať. V základe sa rozlišujú 4 druhy udalostí, ktoré môžu nastať. Preto ak je nechcené aby užívateľ bol vyrušovaný notifikátorom, napríklad pri nejakom prichádzajúcom súbore alebo ostatných veciach, jednoducho ho môže vypnúť.

Môžu to byť:

- Prichádzajúca správa
- Prichádzajúci súbor
- Prichádzajúci odkaz
- Ostatné

Ďalšie čo je možné nastaviť sú výnimky. V prípade, že nasledujúca správa, ktorá užívateľovi prišla je súčasťou už otvoreného okna, môže vypnúť toto upozornovanie, ktoré je aktuálne nežiadúce. Taktiež je možné zvoliť aby notifikátor zareagoval len vtedy, ak otvorené okno nie je v popredí.

Počas hrania alebo pozerania filmu, keď má užívateľ na celej ploche obrazovky niečo zobrazené sa hodí Múd celej obrazovky. To znamená, že sa v tomto prípade môže nastaviť, či ho bude notifikátor vyrušovať alebo nie. V prípade, že je počítač uzamknutý, je možné vypnúť alebo zapnúť funkciu blikania aj pri tejto udalosti. Výnimky:

- Notifikuj, ak je správa otvorená
 - a súčasne nie je na pozadí
- Notifikuj, ak je počítač uzamknutý
- Notifikuj, ak je režim celej obrazovky

V prípade, že je ako druh notifikácie nastavené blikanie, dá sa nastaviť počet a rýchlosť bliknutí, pomocou posuvníkov, kde u každého je zobrazená nastavená hodnota.

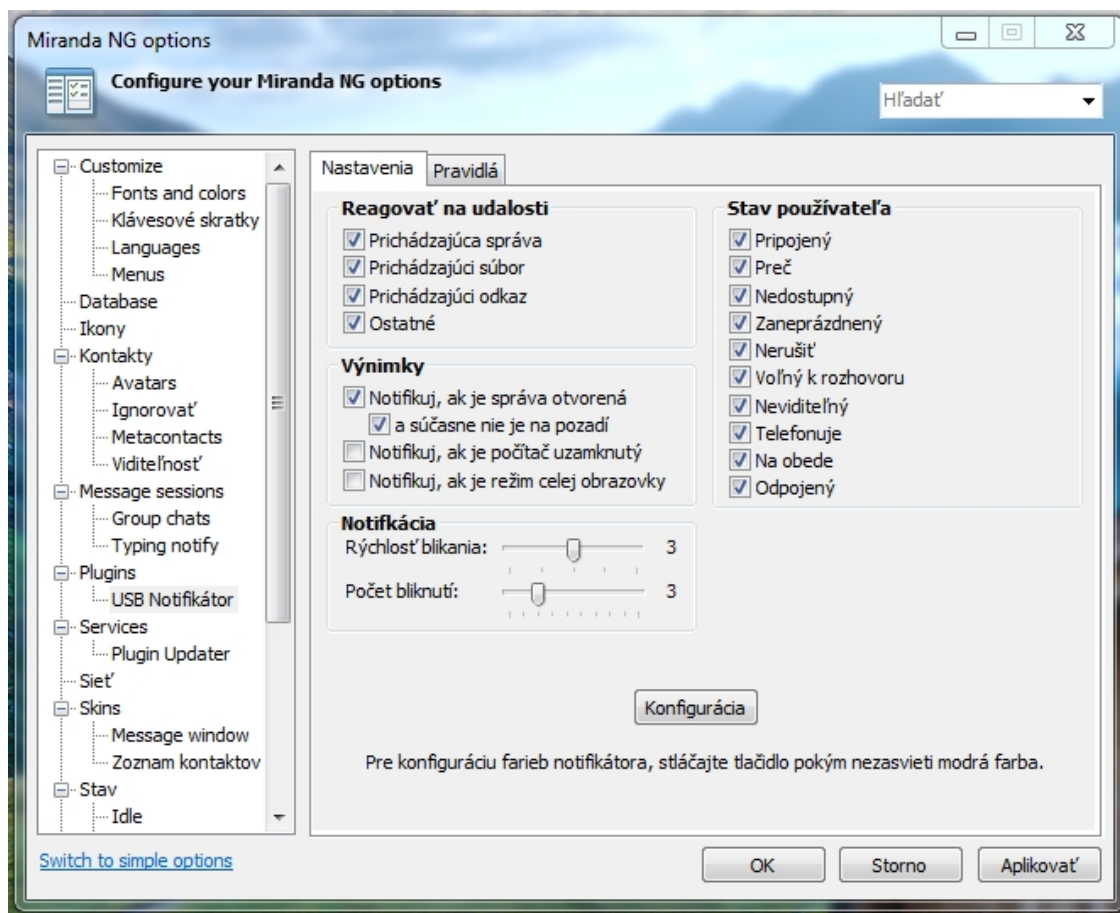
- Rýchlosť blikania
- Počet bliknutí

V klientovi je možné nastaviť viacero druhov statusov pre stav, ktorý je aktuálne vyžadovaný. Preto ak je užívateľ napríklad preč alebo nedostupný nie je žiadané aby notifikátor reagoval. To, či to je povolené, alebo nie, je možné samozrejme zmeniť podľa požadovaných preferencií. V ďalšej kolónke s nastaveniami je možné jednotlivito definovať správanie notifikátora pri hore spomenutých udalostiach, na ktoré zareagovať. Notifikuj keď je status nastavený na:

- Pripojený
- Preč
- Nedostupný
- Zaneprázdnený
- Nerušiť
- Voľný k rozhovoru
- Neviditeľný
- Telefonuje
- Na obede
- Odpojený

Posledným objektom v tejto záložke je Konfiguračné tlačidlo slúžiace na nastavenie notifikátora. Je to z toho dôvodu, kedy sa počas testovania, nastavené adresy farieb notifikátora, na určitú dobu pozmenili. Taktiež je potrebné pri nastavení inej verzie

notifikátora, než bol použitý pri programovaní.



Obr. 4.1: Uživatelské prostredie - Nastavenia

4.1.2 Pravidlá

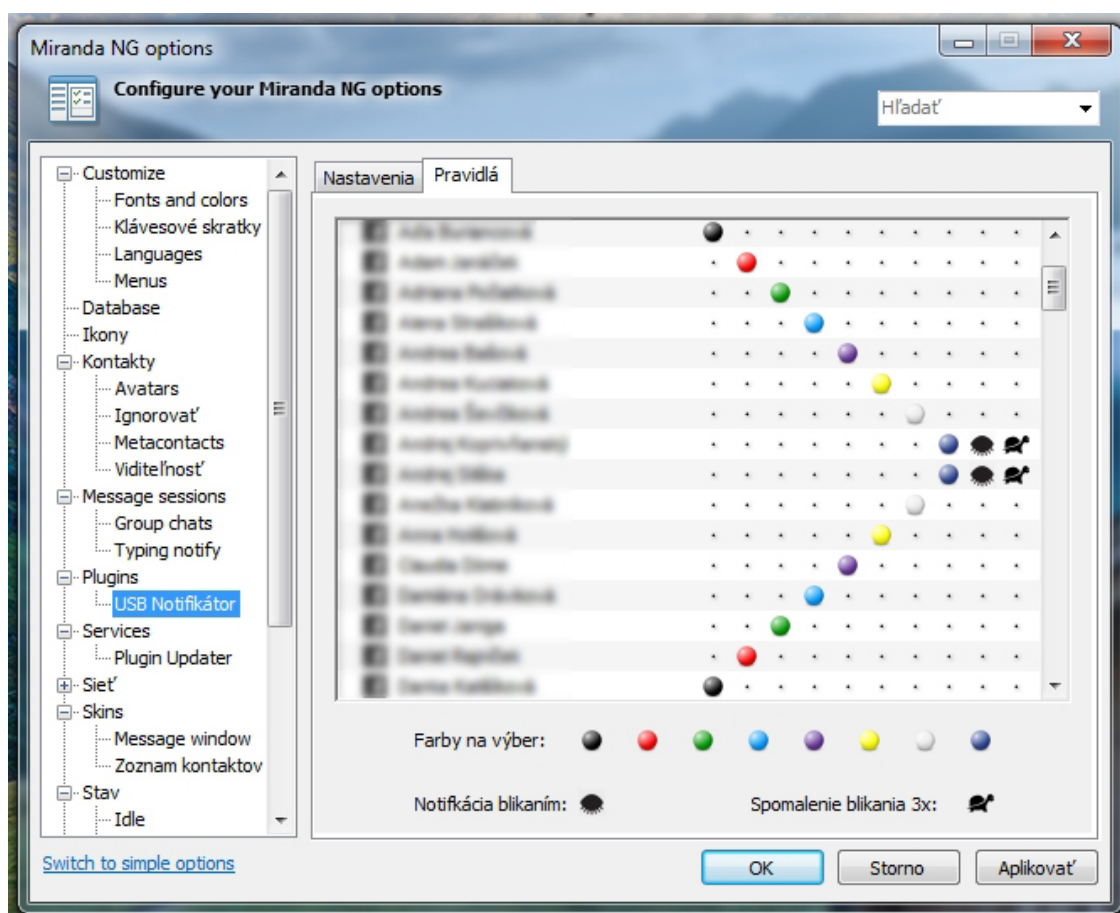
Druhá záložka obsahuje rozsiahli zoznam všetkých vašich kontaktov, zo všetkých pridaných účtov v Mirande NG. Každému z nich sa dá nastaviť farba notifikácie, zapnúť notifikácie pomocou blikania, kedy v základe je nastavené svietenie, ktoré trvá až do otvorenia okna so správou, nastaviť spomalené blikanie, alebo notifikácie vypnúť úplne.

Farba notifikátora:

- Čierna = Vypnuté
- Červená
- Zelená
- Tyrkysová

- Fialová
- Žltá
- Biela
- Modrá

Zvolený notifikátor má v sebe už vopred nakonfigurovaných sedem druhov farieb, ktoré sa nedajú meniť. Preto sú tieto farby možno trochu nelogicky zvolené, avšak je ich dostatočný počet na to, aby pokryli svojim množstvom rôzne druhy notifikácií.



Obr. 4.2: Uživatelské prostredie - Pravidlá

Po výbere hardvérového zariadenia, určení definovaných požiadaviek na funkcie, vzhľadu a celkového dizajnu, sa môže začať programovať softvérová časť - plugin, ktorá je popísaná nižšie.

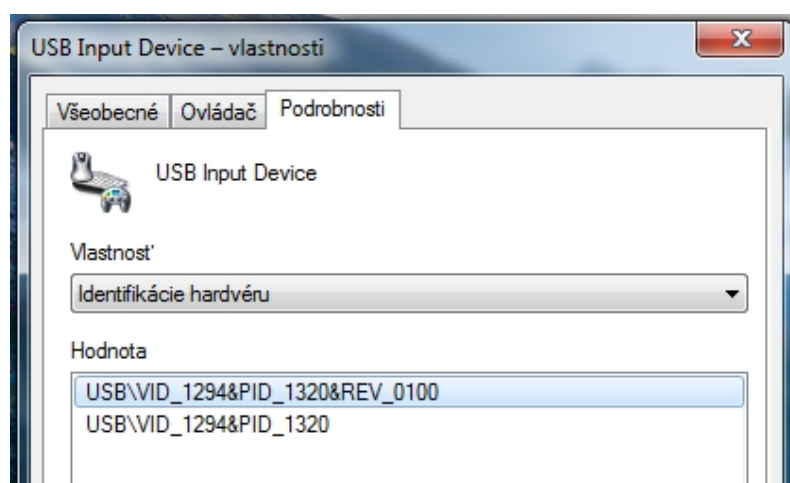
5 VÝVOJ A TVORBA PLUGINU

5.1 Popis komunikácie s notifikátorom

Na internete je viacero už hotových knižníc, ktoré dokážu zabezpečiť komunikáciu so zariadeniami so štandardom HID. Výhodou pre túto prácu je aj to, že komunikáciu medzi počítačom a notifikátorom zabezpečuje už naprogramovaná knižnica od Simona Innsa, ktorá podstatne zjednodušuje prácu so zariadením. Knižnica je voľne dostupná pod licenciou GPL, čo znamená, že ju je možné bez obmedzení kopírovať, meniť alebo distribuovať. Po importovaní knižnice do tohto projektu vo Visual Studiu je možné volať jednotlivé funkcie. Podrobnejšie informácie o tom ako knižnica funguje sú spomenuté v článku autora. [6]

5.1.1 PID a VID Identifikátory

Každé zariadenie, ktoré je pripojené k USB portu je identifikované pomocou unikátnej kombinácie PID a VID identifikátorov. Sú to vlastne dve šesťnásť bitové čísla, slúžiace k nájdeniu príslušných ovládačov v počítači.[3] Čiže na základe nich pri pripojení hocijakého zariadenia k počítaču sa najskôr overí, či zariadenie má dostupné potrebné ovládače. Ak má, tak pri prvom pripojení pridá zariadenie do zoznamu nainštalovaných a ak nie ponúkne dodatočné nainštalovanie.



Obr. 5.1: Identifikátory PID a VID

V prvom rade treba zistiť identifikátory PID a VID notifikátoru. Tie sa dajú jednoducho zistiť v Správcovi zariadení v operačnom systéme Windows. Na začiatku

komunikácie sa zavolá funkcia „findDevice“ s dvoma parametrami, čo sú v našom prípade identifikátory, ktorá určí, do ktorého zariadenia v počítači sa budú nasledujúce príkazy zasielať. Následne pomocou funkcie „sendUsbCommandWriteOnly“ je možné odosielať požadované príkazy. Sú to v skutočnosti čísla od 0 až po 7. Nula znamená vypnuté zariadenie a ostatné čísla sú farby podľa tabuľky príkazov.

Príkaz	Farba
0	vypnuté
1	červená
2	zelená
3	tyrkysová
4	fialová
5	žltá
6	bledo modrá
7	modrá

Tab. 5.1: Tabuľka príkazov

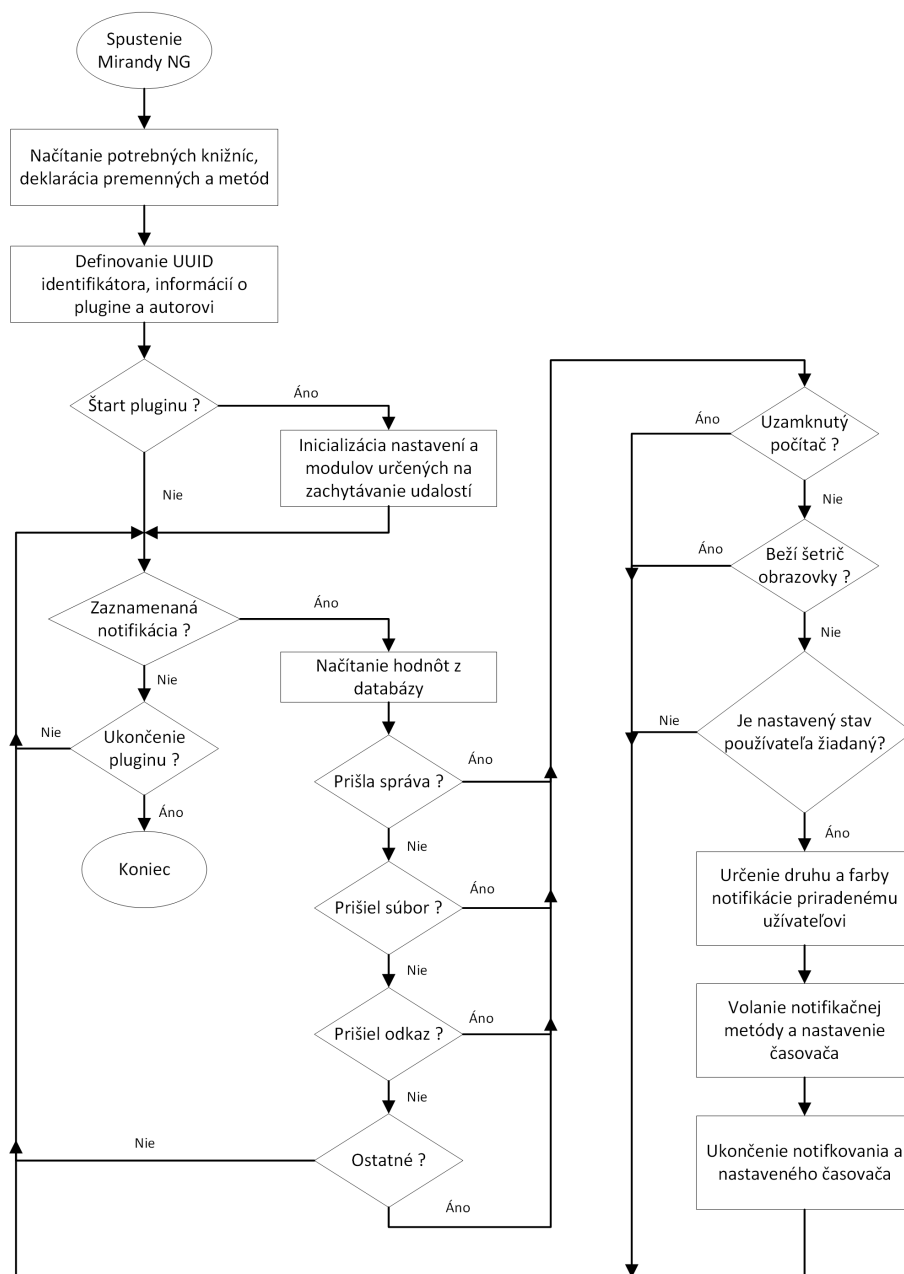
5.2 Zaznamenanie udalostí

Princíp fungovania pluginov ako aj celého klienta Miranda NG je uskutočňovaný na základe určitých udalostí. Sú vykonávané v určitom slede podľa toho, ktorá časť programu ho vyvolá. Môže to byť buď užívateľ, alebo niektorá časť programu, ako napríklad notifikácie alebo nejaký druh načasovania udalosti.

V mojom prípade, v základe plugin rozlišuje tri druhy udalostí, a to je udalosť Inicializácie nastavení, ktorá po spustení klienta načíta všetky najskôr preddefinované, a pokiaľ ich nejako užívateľ pozmenil tak aj ním uložené nastavenia. Druhý typ udalostí, ktorá môže nastať je Prichádzajúca notifikácia. Tá sa vykonáva pokiaľ užívateľovi príde správa, súbor alebo nejaký URL odkaz. Posledným typom je udalosť Otvorenie notifikačného okna prichádzajúcej správy, kedy sa notifikátor vypne. Toto členenie súčasne určuje aj to, či na konkrétny druh notifikácie má notifikátor zareagovať.

Ďalšou súčasťou pri programovaní pluginu sú takzvané „flagy“. Sú vopred definované v API Mirandy NG a využívajú sa na zistenie jednotlivých zmien v programe,

ktoré môžu nastať. Pri zavolaní metódy s určitým flagom je tak možno určiť napríklad stav dostupnosti používateľa online, získanie bližších informácií z jeho profilu, alebo čohokoľvek iného.



Obr. 5.2: Vývojový diagram

5.3 Popis zdrojových kódov pluginu

Kedže z SVN je možné stiahnuť testovaciu verziu pluginu, ktorá obsahuje už preddefinované základné hlavičkové súbory, telička programu a základnú stavbu potrebnú pre pridanie pluginu do Mirandy NG, nebol tento projekt založený ako nový, ale je na tomto už vytvorenom založený. Po stiahnutí všetkých potrebných súborov je následne potrebné namapovanie všetkých dodatočných knižníc a zdrojových kódov, s ktorými bude Visual Studio následne pracovať.

V prvom rade je definovaná štruktúra pluginInfo obsahujúca informácie o autorovi, jeho kontakt, krátky popis programu a jedinečný identifikátor UUID. Hlavnú časť pluginu potom tvoria metódy. Sú to buď vopred definované metódy využívajúce pridané knižnice, alebo mnou novo vytvorené, ktoré sú obsiahnuté v súbore main.cpp. Môžeme ich rozdeliť na dve najzákladnejšie:

- Load() - Stará sa o to, čo má plugin vykonať pri jeho štarte
- Unload() - Stará sa o to, čo má plugin vykonať pred jeho ukončením

Unload nebola v tomto projekte použitá. Metóda Load obsahuje takzvané odchytenie udalostí alebo „HookEvent“. Obsahuje dva parametre. Prvý z nich, definuje typ zachytenej udalosti, druhý z nich definuje čo sa má vykonať pri danej udalosti.

- ME_OPT_INITIALISE - Inicializácia nastavení
- ME_DB_EVENT_ADDED - Odchytenie prichádzajúcej správy

Inicializácia nastavení obsahuje základne vlastnosti pluginu ako sú:

- Určenie, ktorý dialóg sa zobrazí ako prvý
- Názov pluginu zobrazeného v menu Mirandy NG
- Umiestnenie pluginu v menu Mirandy NG

Plugin je rozdelený do troch dialógov označených vlastným ID, z ktorého každý obsahuje grafické prvky, taktiež majúce vlastné ID, obsahujúce nadefinované svoje správanie a funkcie, ktoré má vykonávať. Prvky obsahujú, podobne ako aj celý plugin, Inicializačné a Notifikačné časti plus navyše Príkazovú časť. V inicializačnej časti dialógov je určené miesto, kde sa majú zobrazíť zadané súčasti pluginu v menu klienta.

- WM_INITDIALOG - inicializácia

- WM_NOTIFY - notifikácia
- WM_COMMAND - príkazy

V prvom rade sa z databázy pluginu označenej USB_NOTIFIKATOR načítajú dáta do jednotlivých premenných, ktorá každá z nich má svoje jednoznačné ID pod, ktorým je v databáze uložená. Následne sa prvky nastavlia do určených stavov. V prípade, že užívateľ niečo pozmení, program to zaznamená v notifikačnej časti a po následnom potvrdení nastavenia uloží. Príkazová časť následne dané nastavenia vykoná.

- db_get_b - načítanie bajtu z databázy
- db_set_b - uloženie bajtu do databázy
- db_get_w - načítanie wordu z databázy
- db_set_w - uloženie wordu do databázy

5.3.1 Princíp notifikácie

Počas notifikovania je potrebné, aby samotná notifikácia bežala samostatne, nezávisle od ostatných častí Mirandy NG, na pozadí. Základná metóda Sleep() je preto nevhodná, keďže pozastavuje beh celej Mirandy NG a správa by v tomto prípade prišla až po doblikaní. Na toto sú v Mirande NG určené časovače.

- SetTimer - Spustenie časovača s zadefinovaným časom
- KillTimer - Ukončenie časovača

Celkovo sú v plugine využité dva časovače. Jeden z nich opakovane každú sekundu overuje to, či okno so správami nebolo otvorené a v prípade, že áno, všetky notifikácie zruší a notifikátor vypne. Druhý z časovačov sa využíva pri blikaní notifikátora, kedy vytvára časové prestávky medzi bliknutiami.

5.3.2 Použité zdrojové kódy

V projekte sa použilo niekoľko voľne šíriteľných zdrojových kódov, ktoré sú obsiahnuté v ostatných pluginoch a boli modifikované pre potreby tohto pluginu. Je to hlavne z dôvodu zjednodušenia si práce, kedy vytváranie metód s rovnakou alebo podobnou funkčnosťou by nemalo zmysel. Umiestnené sú v repozitári SVN.[15]

5.3.3 Metódy starajúce sa o výslednú notifikáciu

Pre fungovanie pluginu, bolo potrebné naprogramovanie nových doplnkových metód. Prvou z nich je metóda Ignorovať určená na získanie nastavenej farby pre notifikovanie určitého používateľa majúca dva parametre. Prvý parameter z nich určuje používateľa a druhý overuje nastavenú farbu. Pri zhode jednej z farieb vráti hodnotu TRUE.

Následne sa volá metóda Notifikuj s ID príslušnej farby, kedy ku zhode došlo. Táto metóda má na starosti získanie dodatočných informácií z databázy, ako je to či má notifikátor svietiť, blikať a ak áno tak jeho počet a rýchlosť bliknutí. Tieto informácie sú následne spracované súhrnom podmienok.

Metóda Blikaj vykonáva výslednú notifikáciu, spúšťa časovač so zadanými požiadavkami a taktiež dáva pozor na to, keby mohli do konfliktu prísť dve notifikácie. Toto rieši jednoducho vypnutím časovača a následným prehodením situácie podľa nadefinovaných prednostných pravidiel.

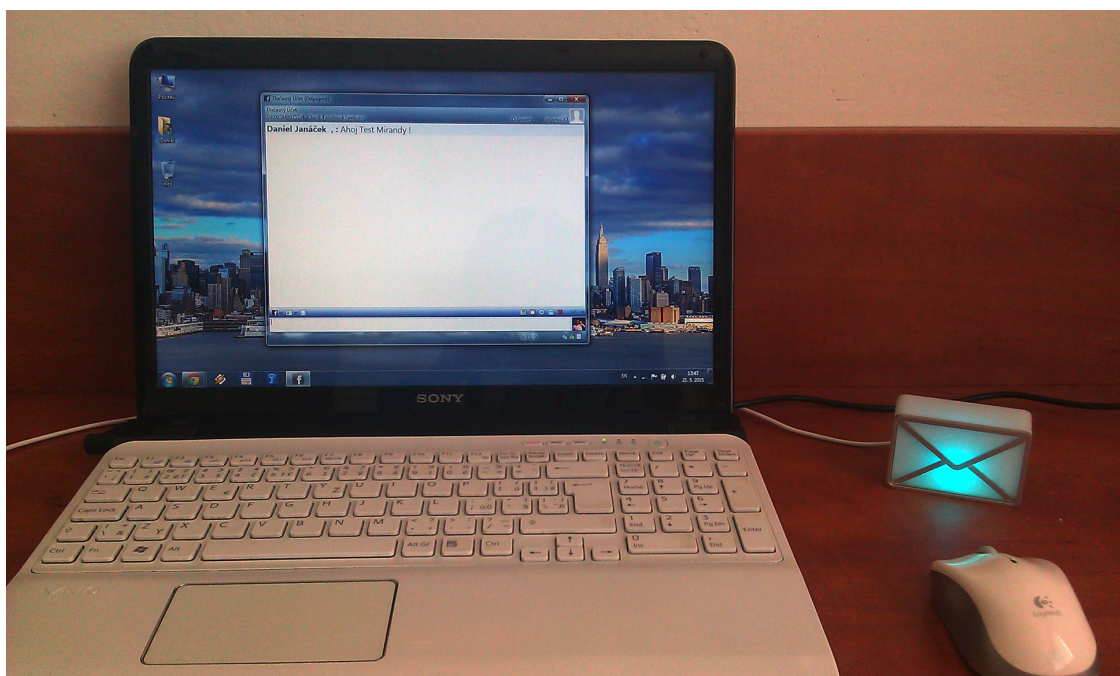
Po vytvorení celého pluginu, je možné prejsť ku konečnému testovaniu spolupráce zariadenia a softvéru v nasledujúcej kapitole.

6 TESTOVANIE PLUGINU

Keďže bolo potrebné nejakým spôsobom overiť funkčnosť notifikátora, prebiehalo pred dokončením pluginu testovanie hneď dvoma spôsobmi. A to jednak počas programovania a neskôr po naprogramovaní všetkých funkcií, celková funkčnosť pluginu.

Prvé testovanie prebiehalo tak, že sa pre tieto účely museli založiť nové účty na sociálnych sieťach a jabberi, z ktorých sa posielali správy a súbory, a tak overovalo, či notifikátor zareaguje. Počas ladenia pluginu bolo potrebné niekoľko krát opakovane vypínať, skompilovať projekt, a následne znova pustiť Mirandu NG, aby sa zmeny prejavili.

Testovalo sa hlavne čo najviac rôznych možných kombinácií, kedy by mohlo mať zariadenie problémy. Boli to napríklad notifikácie od viacerých používateľov súčasne, to či dokáže prerušiť jeden druh notifikácie a pokračovať v nej po skončení predchádzajúcej, a to hlavne vtedy, ak notifikátor svieti, súčasne do neho príde iná úloha ako je blikanie a následne sa sledovalo jeho vysporiadanie s týmito ako aj inými stavmi. Taktiež správne nulovanie premenných, časovačov a ich počiatočné alebo nastavené hodnoty.



Obr. 6.1: Test notifikátora

Neskôr počas niekoľkých dní prebiehal druhý test, kedy po odladení zistených

chýb, bežal na pozadí v Mirande NG hneď po spustení počítača. Vtedy sa odladili posledné nedostatky a plugin je tak považovaný za hotový. Aj keď z pohľadu tejto práce sa jedná o finálnu verziu, je možné, že časom a množstvom potencionálnych používateľov tohto pluginu sa nájdu ďalšie nedostatky, ktoré nemuseli byť hneď na prvý pohľad viditeľné, pretože každý používateľ môže tento plugin používať s rôznymi kombináciami nastavení, doplnkov a tém.

7 ZÁVER

V tejto bakalárskej práci som sa najskôr oboznámil so štandardom HID, popísal jeho vlastnosti, funkcie a jeho využitie v štandarde USB. Spravil som rešerš dostupných zariadení, ktoré by boli teoreticky schopné notifikovať správy alebo emaily z klienta Miranda NG. Počas hľadania sa mi podarilo nájsť viacero druhov riešení, avšak na základe vybraných požiadaviek, medzi ktoré patrili možnosť pripojenia do USB portu, možnosť implementácie do klienta Miranda NG a dostupná cena, bolo zo všetkých spomenutých iba jedno vyhovujúce. Je to Dream Cheeky 815 Webmail Notifier.

Následne som sa oboznámil so zdrojovými kódmi programu Miranda NG a navrhol som dizajn užívateľského prostredia pluginu, ktorý je čo najviac prehľadný, jednoduchý, intuitívny a hlavne jednoducho ovládateľný pre každého potenciálneho používateľa.

Neskôr som našiel, naštudoval a implementoval do prostredia Visual Studio 2013 voľne dostupnú knižnicu od Simona Innsa, ktorý sa podobnou problematikou komunikácie s externými zariadeniami typu HID už zaoberal a podstatne mi následne uľahčil prácu pri programovaní a komunikácií medzi notifikátorom a klientom Miranda NG.[3]

V poslednom kroku som prešiel k samotnému programovaniu pluginu, ktorý som po naprogramovaní podrobil testom a doladil do finálnej podoby. Tak ako aj každá iná aplikácia bežiaca pod operačným systémom Windows, potrebuje svoje aktualizácie, bude aj tento plugin mať svoje prípadné opravy. Zdrojové kódy budú po odovzdaní práce umiestnené do SVN repozitára Mirandy NG, kde si ich bude môcť ktokoľvek stiahnuť, upraviť, a možno aj pracovať na ďalšom vývoji tohto pluginu.

LITERATÚRA

- [1] THINGM.COM: *Blink*. [online]. [cit. 2015-01-06]. Dostupné z URL: <<http://blink1.thingm.com/about/>>.
- [2] AMAZON.COM: *Dream Cheeky 815 WebMail Notifier*. [online]. [cit. 2015-01-06]. Dostupné z URL: <<http://www.amazon.com/Dream-Cheeky-815-WebMail-Notifier/dp/B001L23EZ0/>>.
- [3] VOTI.NL: *How to obtain an USB VID/PID for your project*. [online]. [cit. 2015-01-08]. Dostupné z URL: <<http://www.voti.nl/docs/usb-pid.html>>.
- [4] MIRANDA-NG.ORG: *Miranda NG*. [online]. [cit. 2015-01-08]. Dostupné z URL: <<http://www.miranda-ng.org/cs/>>.
- [5] RAZERZONE: *Razer DeathStalker Ultimate*. [online]. [cit. 2015-01-06]. Dostupné z URL: <<http://www.razerzone.com/gaming-keyboards-keypads/razer-deathstalker>>.
- [6] INNS, Simon. *Open Source Visual C++ Class for USB Generic HID Communication*. 2015. [online]. [cit. 2015-01-08]. Dostupné z URL: <http://www.waitingforfriday.com/index.php/Open_Source_Visual_C%2B%2B_Class_for_USB_Generic_HID_Communication>.
- [7] BLUETOOTH.ORG: *HID - Human Interface Device*. [online]. [cit. 2015-05-15]. Dostupné z URL: <[hhttps://developer.bluetooth.org/TechnologyOverview/Pages/HID.aspx](https://developer.bluetooth.org/TechnologyOverview/Pages/HID.aspx)>.
- [8] VISUAL STUDIO 2013: *Program MSDN Academic Alliance*. [online]. [cit. 2015-01-08]. Dostupné z URL: <<https://msdnaa.feec.vutbr.cz/web/>>.
- [9] AMAZON: *SONY PS DualShock 4*. [online]. [cit. 2015-01-06]. Dostupné z URL: <<http://www.amazon.com/DualShock-Wireless-Controller-PlayStation-Camouflage-4/dp/B00KVP78FE>>.
- [10] PLAYSTATION: *SONY PS DualShock 4 Wireless Controller*. [online]. [cit. 2015-01-06]. Dostupné z URL: <<http://www.playstation.com/en-us/explore/accessories/dualshock-4-wireless-controller/>>.
- [11] INSTRUCTABLES.COM: *USB Real-Time Notification Device*. [online]. [cit. 2015-01-06]. Dostupné z URL: <<http://www.instructables.com/id/USB-Real-Time-Notification-Device/>>.

- [12] KEIL USB component: *Human Interface Device*. [online]. [cit. 2015-05-06]. Dostupné z URL: <http://www.keil.com/pack/doc/mw/USB/html/_h_i_d.html>.
- [13] USB.ORG: *Device Class Definition for Human Interface*. [online]. [cit. 2015-05-06]. Dostupné z URL: <http://www.usb.org/developers/hidpage/HID1_11.pdf>.
- [14] SILABS: *HUMAN INTERFACE DEVICE TUTORIAL*. [online]. [cit. 2015-05-06]. Dostupné z URL: <<http://www.silabs.com/Support%20Documents/TechnicalDocs/AN249.pdf>>.
- [15] MIRANDA NG: *Repozitár SVN*. [online]. [cit. 2015-05-06]. Dostupné z URL: <<http://svn.miranda-ng.org/main/trunk/>>.

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

USB	Universal serial bus
FTDI	Future Technology Devices International
API	Application programming interface
GPL	General Public Licence
LED	Light-emitting diode
LCD	Liquid Crystal Display
RGB	Red Green Blue
HID	Human interface device
VID	Vendor ID
PID	Product ID
UUID	Universally unique identifier
MSDNAA	Microsoft Academic Alliance
SVN	Apache Subversion
ROM	Read-only memory

ZOZNAM PRÍLOH

A Priložené CD	41
A.1 Zdrojové kódy pluginu	41

A PRILOŽENÉ CD

A.1 Zdrojové kódy pluginu

Priečinok na CD obsahuje zdrojové kódy pluginu pre klienta Miranda NG, napísané sú v jazyku C++ a ako primárny kompilátor je Visual Studio 2013.